XILINX

XAPP1331 June 6, 2018

Aurora 8B10B for GTY UltraScale+, Zynq UltraScale+ MPSoC and RFSoC

Author: Antonello Di Fresco and Paolo Novellini

Summary

This application note targets the Aurora 8B10B protocol for GTY transceivers in UltraScale+™ devices. A reference design with a customized Aurora IP is provided to support the GTY transceiver when connected to a device supporting only the Aurora 8B10B protocol, such as Artix-7. The reference design is based on Xilinx® LogiCORE™ IP AXI Chip2Chip core.

Reference Design

Two reference designs were created to validate the Aurora 8B10B core with GTY transceivers. To verify interoperability, one reference design implements an AXI Chip2Chip Master and the other reference design implements an AXI Chip2Chip Slave. The AXI Chip2Chip Master is developed for either RFSoC or Virtex® UltraScale+ devices. The AXI Chip2Chip Slave is developed for Artix®-7 devices.

For more information on this IP core, see the *AXI Chip2Chip Product Guide* (PG067) [Ref 1].

Download the reference design files for this application note from the Xilinx website.

Tool Flow and Verification

The following checklist indicates the tool flow and verification procedures used for the provided reference design.

Table 1: **Reference Design Matrix**

Parameter	Description
General	
Developer Name	Xilinx
Target Devices	UltraScale+, Zynq UltraScale+ MPSoC and RFSoC
Source code provided?	Yes
Source code format (if provided)	Verilog
Design uses code or IP from existing reference design, application note, 3rd party or Vivado software? If yes, list.	AXI Chip2Chip v5.0 Aurora 8B/10B v11.1 Aurora_8b10b_gty v1.0
Simulation	
Functional simulation performed	Yes

Table 1: **Reference Design Matrix *(Cont'd)***

Parameter	Description
Timing simulation performed?	No
Testbench provided for functional and timing simulation?	Yes
Test bench format	Verilog
Simulator software and version	Vivado simulator 2018.1 and Modelsim 10.6b
SPICE/IBIS simulations	No
Implementation	
Synthesis software tools/versions used	Vivado tools 2018.1
Implementation software tools/versions used	Vivado tools 2018.1
Static timing analysis performed	Yes
Hardware Verification	
Hardware verified	Yes
Platform used for verification	AC701, ZCU1254, VCU1262

For the Master reference design, the top-level code is `axi_chip2chip_master_exdes.v`. The top-level ports are described in Table 2.

Table 2: **axi_chip2chip_master_exdes.v Port List**

Name	Size	Direction	Description
reset	1	Input	Global system reset
gt_refclk_p	1	Input	Differential input clk to GT. 491.52MHz
gt_refclk_n	1	Input	Differential input clk to GT. 491.52MHz
aurora_rx_p_mas	1	Input	Differential serial GT RX input for lane 0.
aurora_rx_n_mas	1	Input	Differential serial GT RX input for lane 0.
aurora_tx_p_mas	1	Output	Differential serial GT TX output for lane 0.
aurora_tx_n_mas	1	Output	Differential serial GT TX output for lane 0.
pma_init	1	Input	GTY Reset.
start_traffic	1	Input	Start traffic generator test.
t_axi_calib_done_out_mas	1	Output	Asserted when Link Detect FSM is in the SYNC state.
t_axi_calib_error_out_mas	1	Output	Multiple bits are received with errors in the Master or Slave AXI Chip2Chip core.
axi_c2c_link_error_out_mas	1	Output	Asserted when the AXI Chip2Chip Slave core is reset during normal operations.
t_axi_phy_error_out_mas	1	Output	Link Detect FSM failed due to a configuration mismatch of Master and Slave AXI Chip2Chip cores.
lite_error	1	Output	AXI4-Lite interface error.
axi4_error	1	Output	AXI4 interface error.

For the Slave reference design the top-level code is `axi_chip2chip_slave_exdes.v`. The top-level ports are described in Table 3.

Table 3: **axi_chip2chip_slave_exdes.v Port List**

Name	Size	Direction	Description
reset	1	Input	Global system reset
gtxq1_p	1	Input	Differential input clk to GT. 491.52MHz
gtxq1_n	1	Input	Differential input clk to GT. 491.52MHz
aurora_rx_p_slv	1	Input	Differential serial GT RX input for lane 0.
aurora_rx_n_slv	1	Input	Differential serial GT RX input for lane 0.
aurora_tx_p_slv	1	Output	Differential serial GT TX output for lane 0.
aurora_tx_n_slv	1	Output	Differential serial GT TX output for lane 0.
pma_init	1	Input	GTP Reset.
t_axi_calib_done_out_slv	1	Output	Asserted when Link Detect FSM is in the SYNC state.
t_axi_calib_error_out_slv	1	Output	Multiple bits are received with errors in the Master or Slave AXI Chip2Chip core.
t_axi_phy_error_out_slv	1	Output	Link Detect FSM failed due to a configuration mismatch of Master and Slave AXI Chip2Chip cores.
lite_error	1	Output	AXI4-Lite interface error.
axi4_error	1	Output	AXI4 interface error.

Hardware Architecture

Figure 1 shows the demonstration test bench with Aurora interface and the high-level block diagram for each reference design.

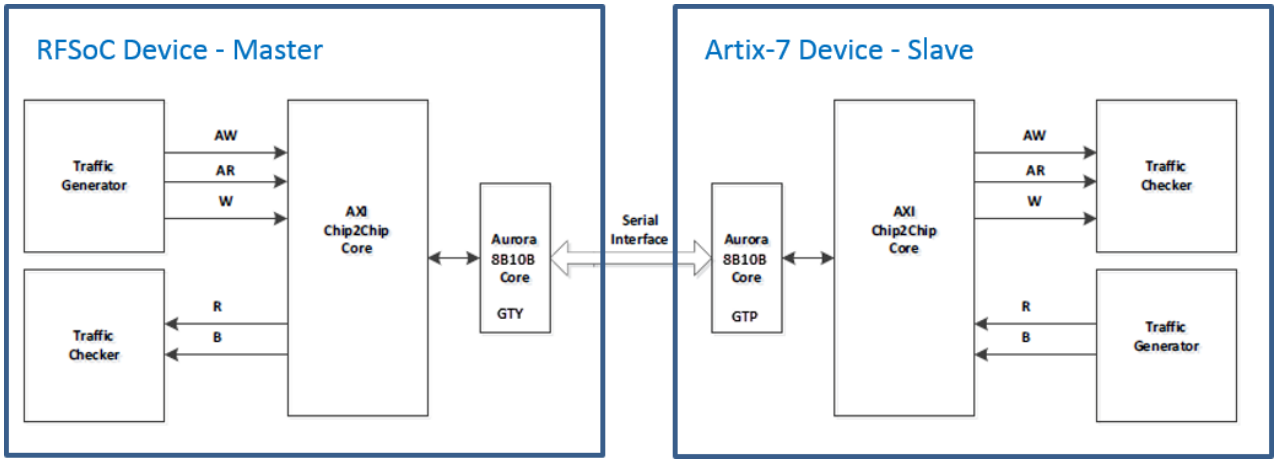

Figure 1: **Hardware Test Bench**

In the Master design, shown in Figure 2, the AXI Chip2Chip and the Aurora 8B10B for GTY are connected using IP Integrator.

The block design is instantiated in a wrapper where there is test logic.

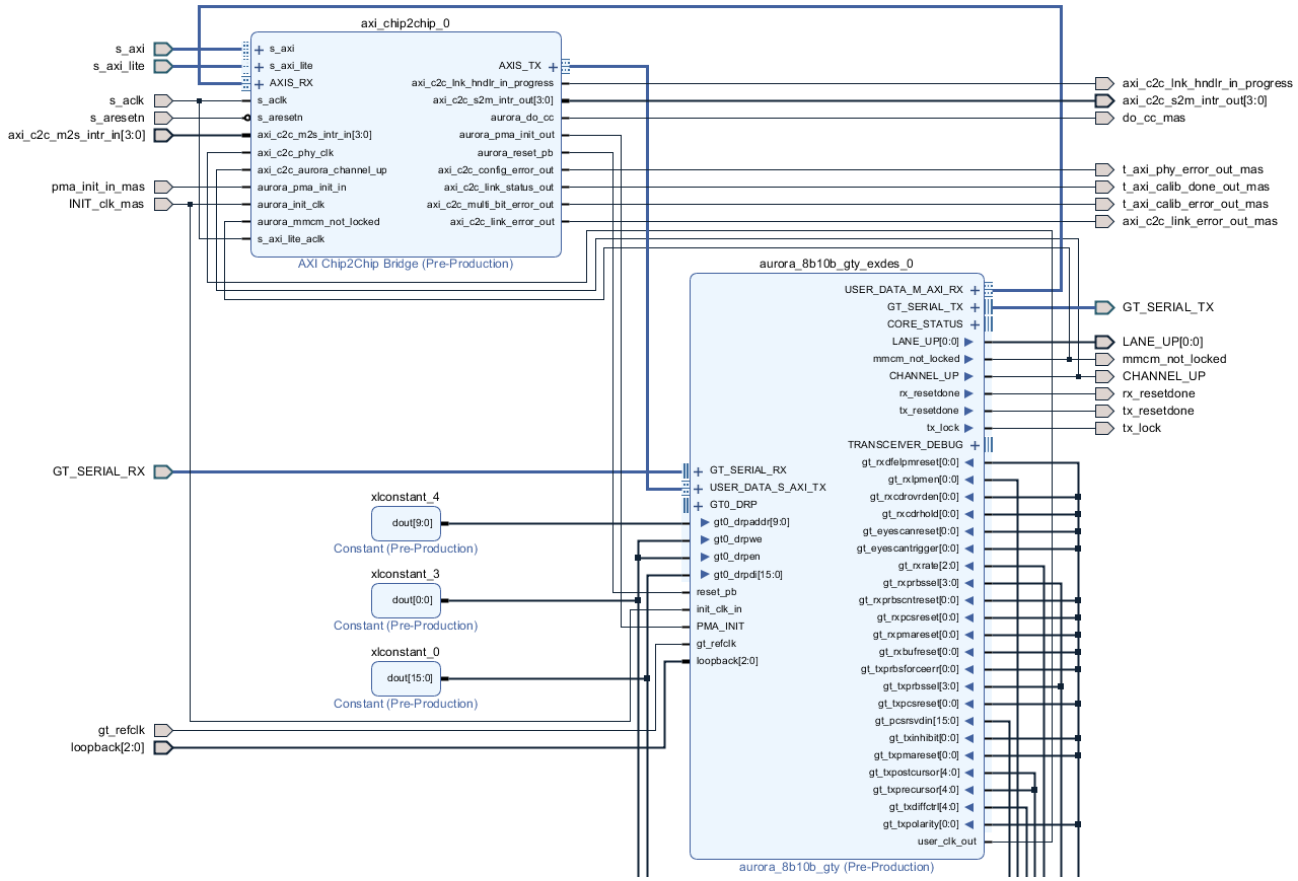

Figure 2: **AXI Chip2Chip and Aurora Block Design**

Figure 3 shows the Master AXI Chip2Chip configuration.

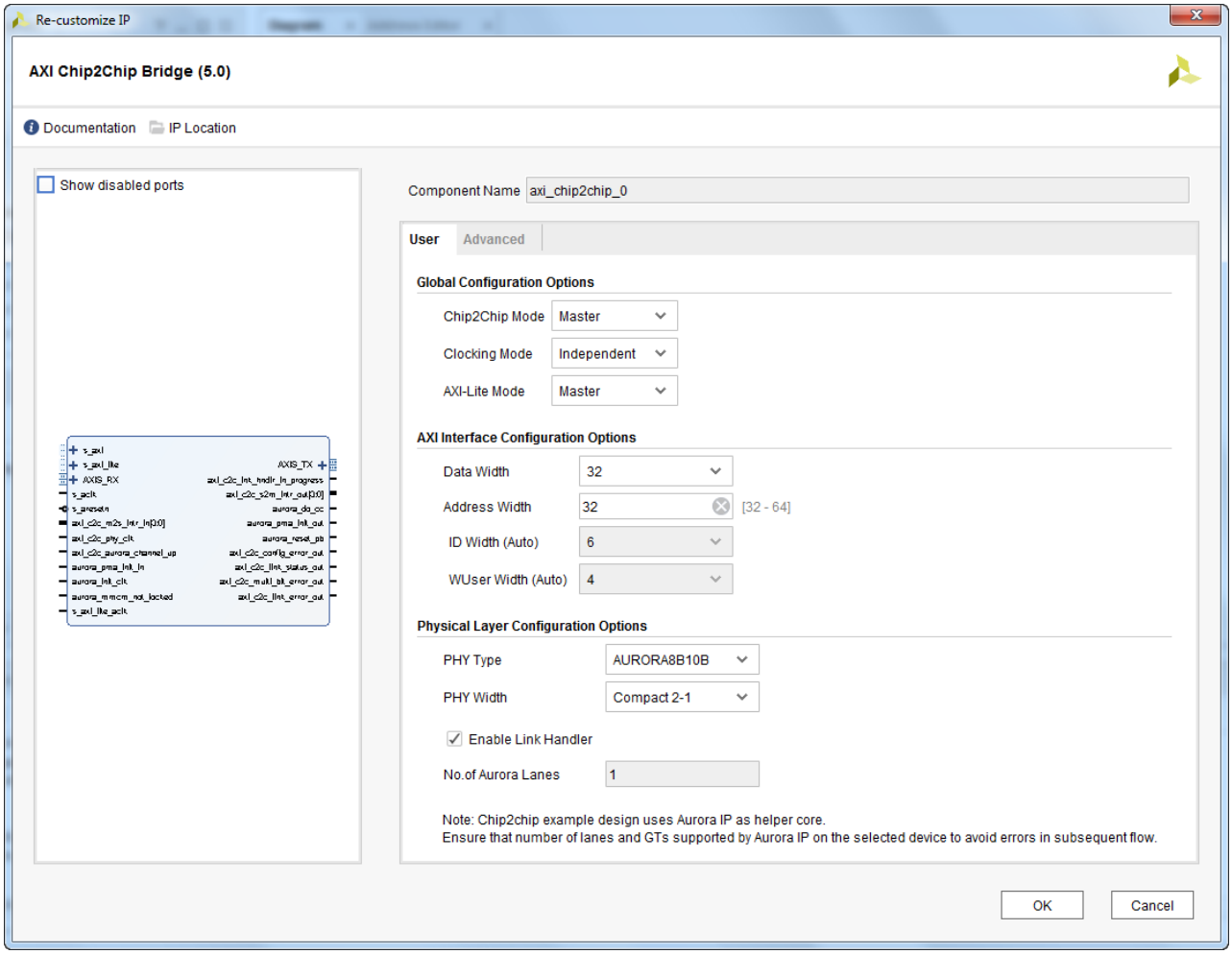

Figure 3: **Master AXI Chip2Chip Configuration**

The final design of the Master side is the same as the example design that is automatically generated from the AXI Chip2Chip core.

For the Slave side, the design has been generated automatically from the AXI Chip2Chip core and configured as Slave.

Figure 4 shows the Slave AXI Chip2Chip configuration.

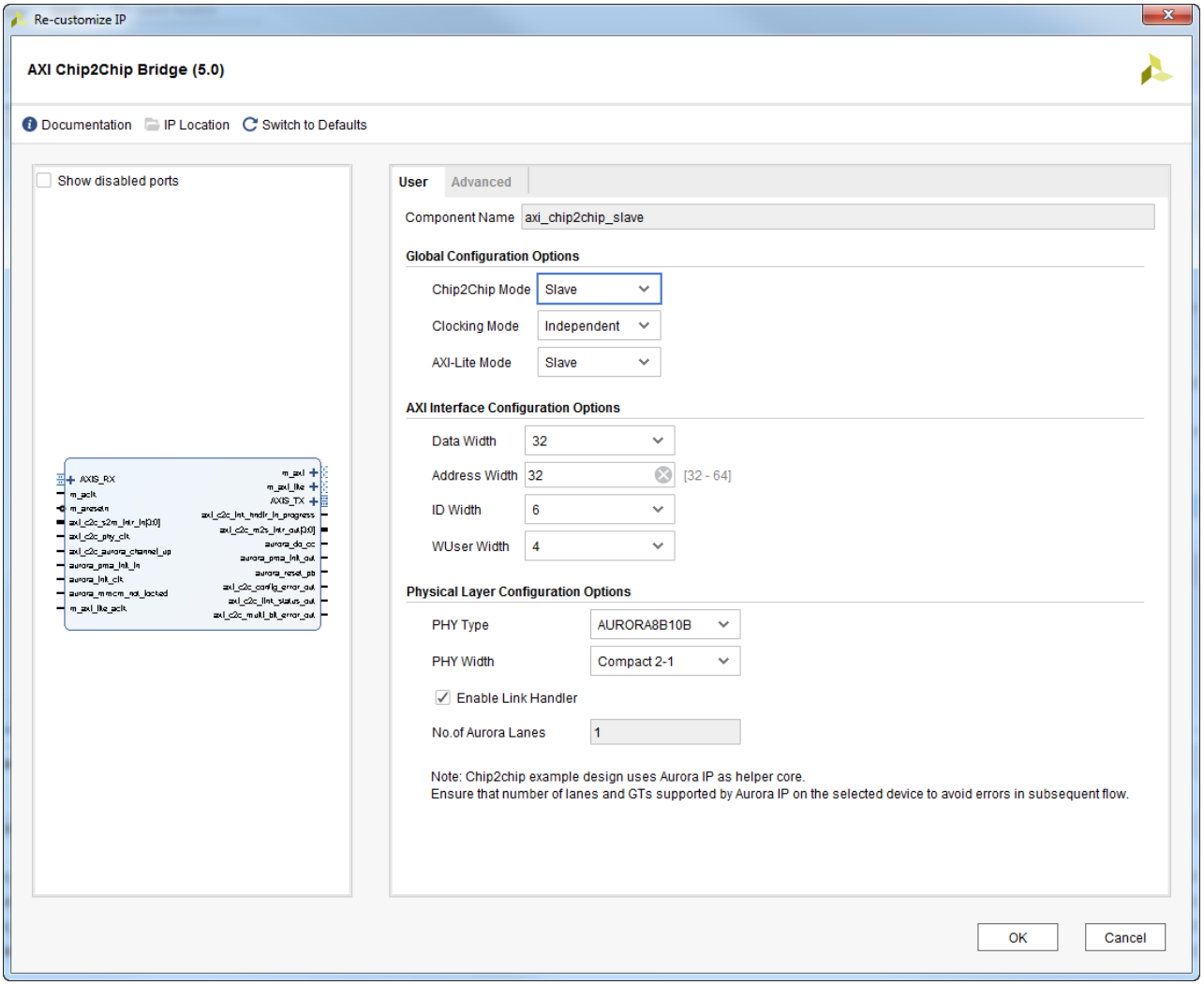

Figure 4: **Slave AXI Chip2Chip Configuration**

Because the line rate cannot be configured from the AXI Chip2Chip wizard, the Aurora core must be reconfigured to be the same line rate of the Master side after the example design Slave Chip2Chip is generated. Figure 5 shows the selected configuration.

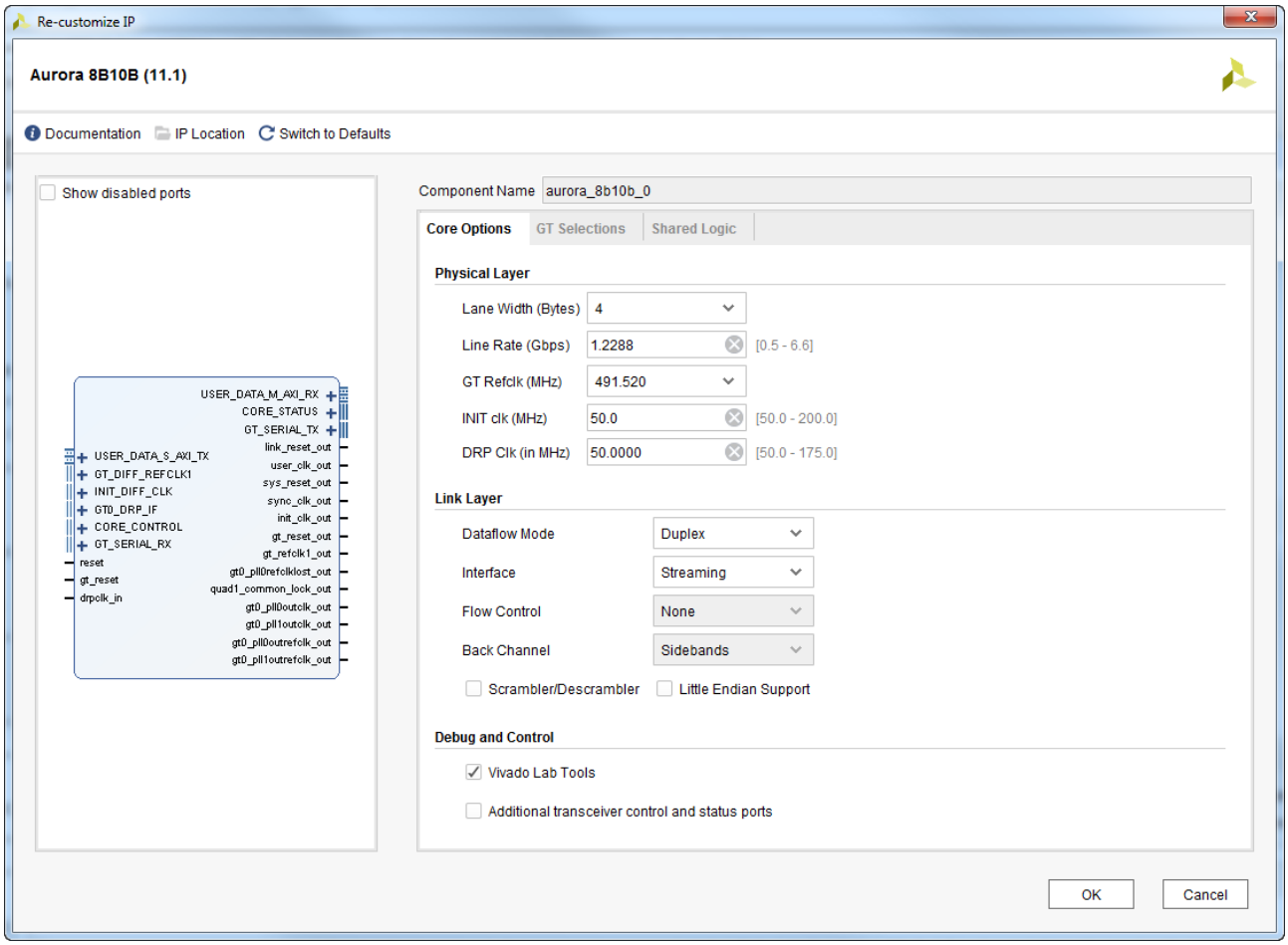

Figure 5: **Aurora 8B10B Slave Configuration**

Clocking Architecture

Table 4 lists the primary clocks required for the design.

Table 4: **Clock Requirements**

Clock	Master (MHz)	Slave (MHz)
Reference Clock	491.52	491.52
AXI	100	100
System	30	50
DRP	30	50

The two reference design require a single external clock of 491.52MHz. The other clocks are generated internally by using a MMCM.

For the master design using an UltraScale+ device, it is possible to connect a MMCM to the input buffer of the transceiver reference clock IBUFDS_GTE4 through a BUFG_GT. The IBUFDS_GTE4 has an optional output ODIV2 to bring the reference clock to the fabric logic. This output can be configured to produce either the O signal or a divide-by-2 version of the O signal. In this design, it is the same frequency of the O signal, or 491.52MHz.

Figure 6 shows the connections of the buffers to the MMCM. The BUF_GT_SYNC is automatically inserted from the Vivado IDE and is not required to be added in the code.

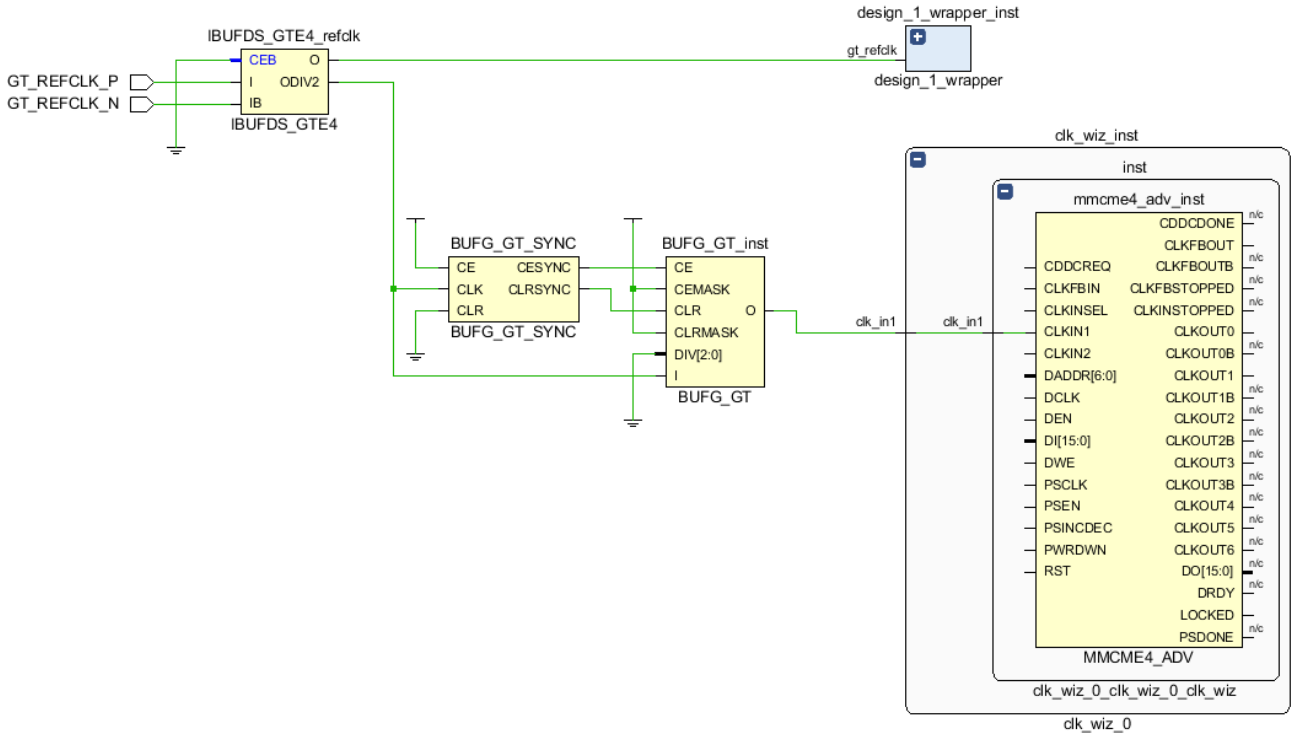

Figure 6: **BUFG_GT to MMCM Connection**

For the Slave design with a 7 series device, it is possible to use a MMCM to generate the required clock from the external reference clock. In this case, the MMCM can be connected directly to the port O of the IBUFDS_GTE2, which is shared with the clock connected to the transceiver.

Figure 7 shows the connections between the IBUFDS_GTE2 and the MMCM.

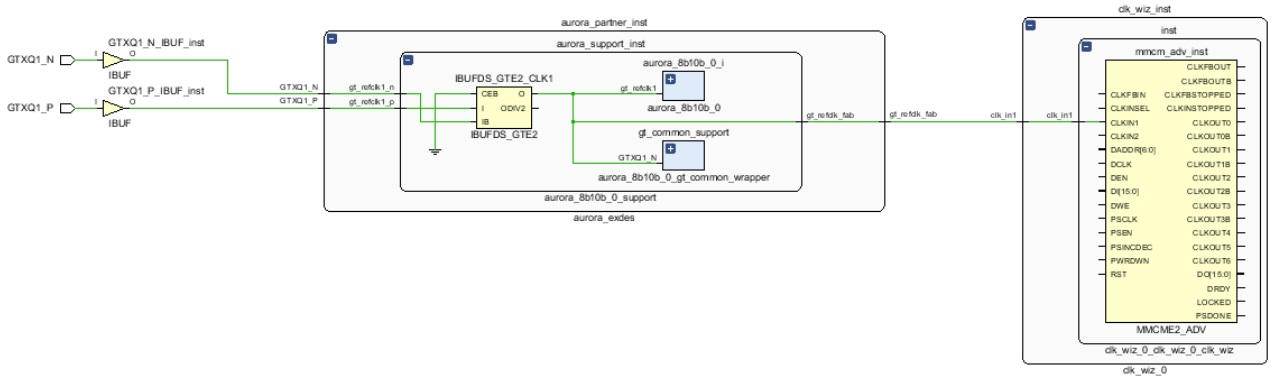

Figure 7: **IBUFDS_GTE2 to MMCM Connection**

Aurora 8B10B GTY

The Aurora 8B10B for GTY is derived from the latest Aurora core v11.1 (which supports GTH and 8b01b) and packaged with the Vivado® IDE. The Aurora core configuration is shown in Figure 8.

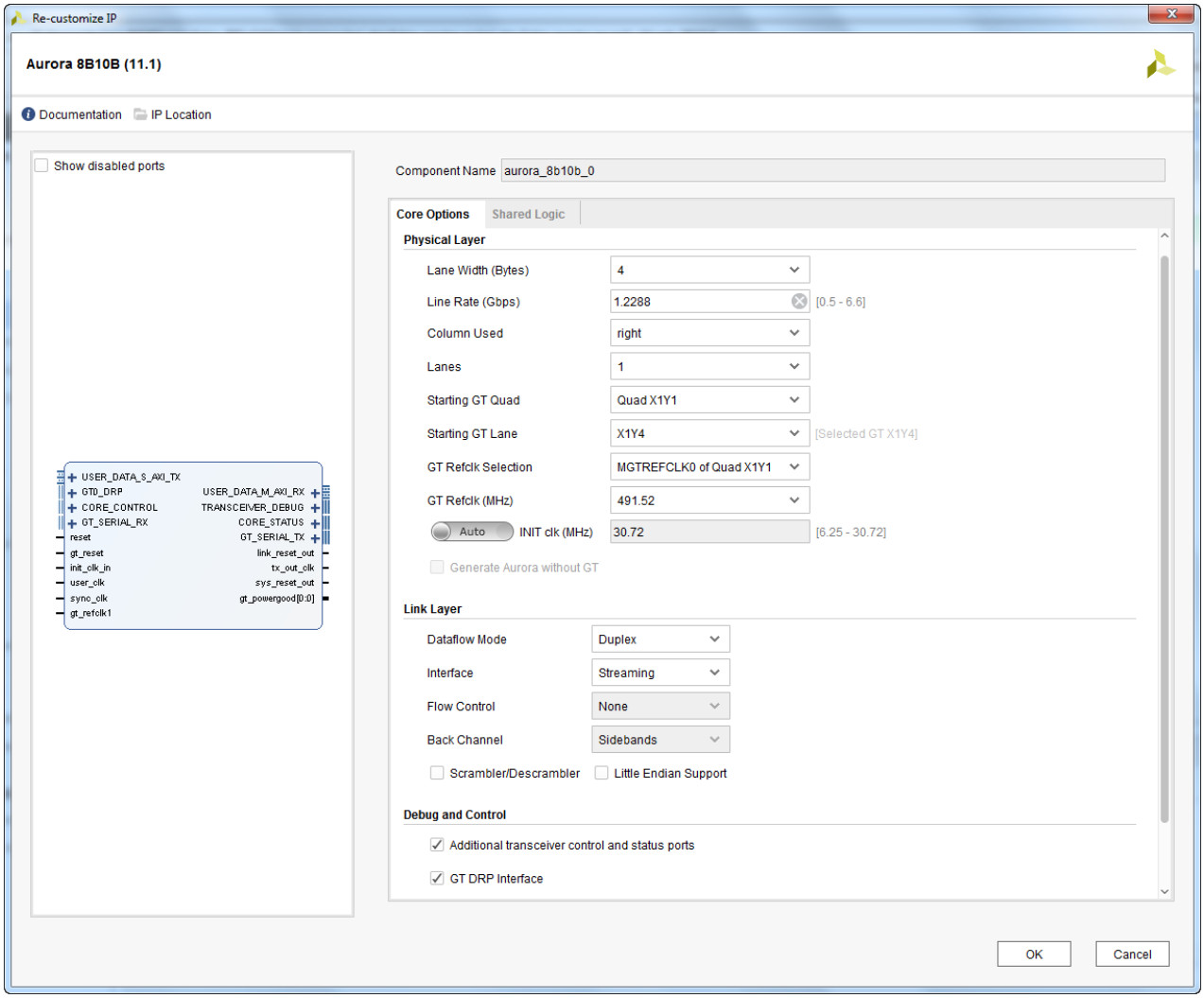

Figure 8: **Aurora 8B10B Configuration**

The xci file defining the GTH is then overwritten with the xci file of the GT wizard that configures the GTY, as shown in Figure 9.

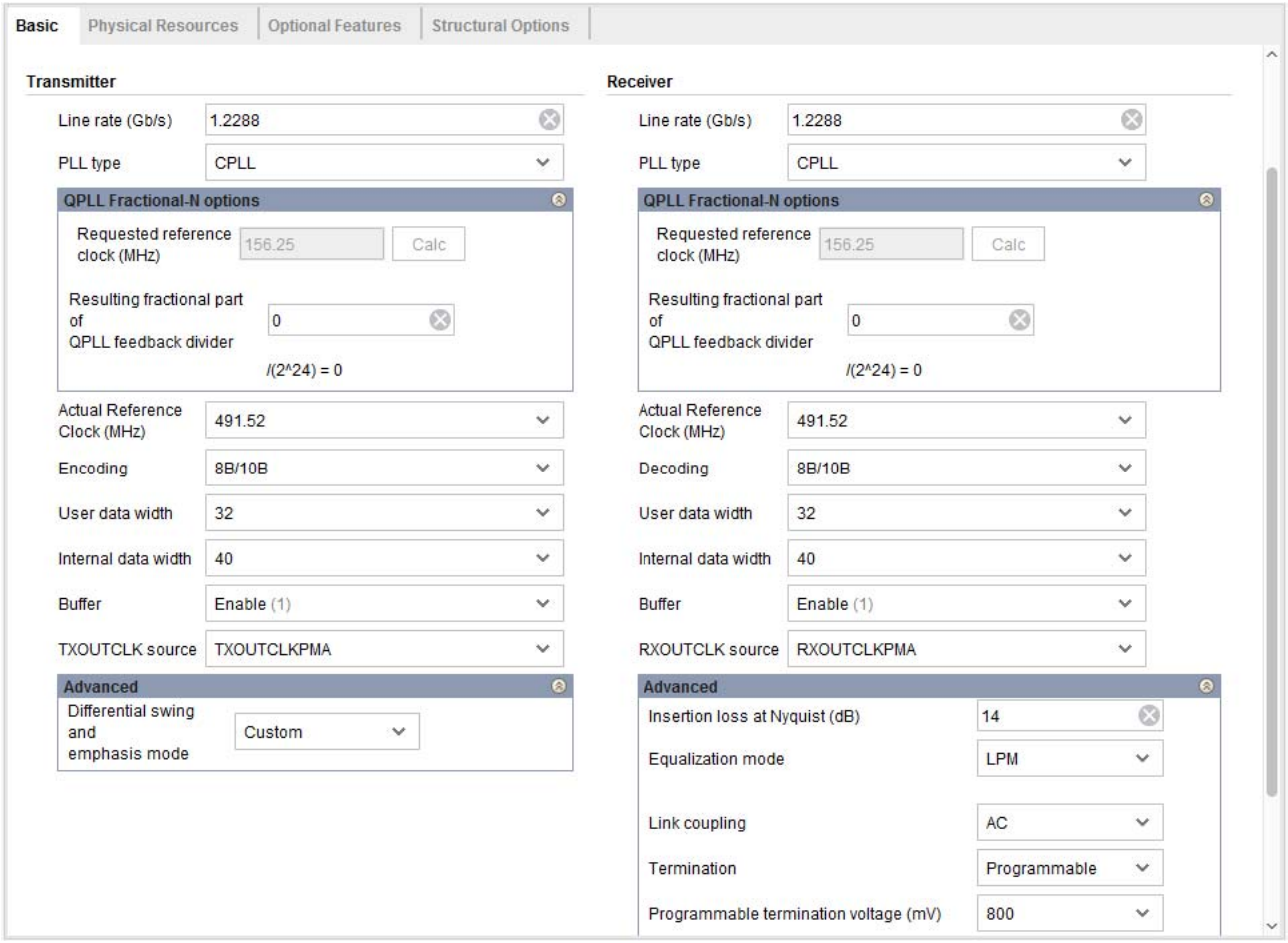

Figure 9: **GTY Configuration**

The Aurora 8B10B for GTY is released as an IP and it is compatible with Vivado IP Catalog and IP Integrator. The IP is called `aurora_8b10b_gty` and it is released as version v1.0 rev.2. Figure 10 shows the symbol of the IP with all available ports.

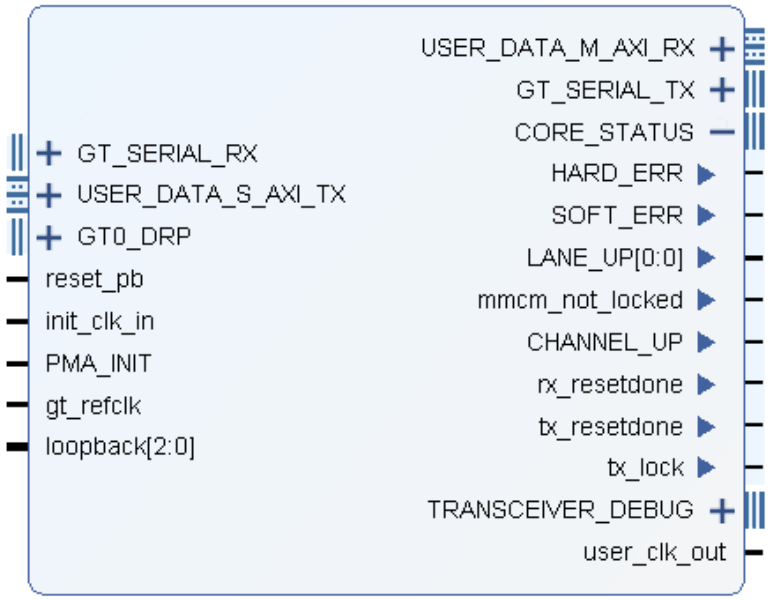

Figure 10: **Aurora 8B10B Symbol**

The Aurora 8B10B ports are listed in Table 5.

Table 5: **Aurora 8B10B Ports List**

Port	Direction	Description
GT_SERIAL_RX	Input	Differential serial data input pin.
gt_refclk	Input	Transceiver Reference Clock. To be connected to an IBUFDS_GTE4.
USER_DATA_S_AXI_TX	Input/Output	User Slave AXI Stream port.
GT0_DRP	Input	DRP port. DRP clock is 30 MHz and can be connected together to intit_clk_in.
reset_pb	Input	Resets the Aurora 8B/10B core (active-High).
init_clk_in	Input	Core clock. 30 MHz.
PMA_INIT	Input	Transceiver reset.
loopback[2:0]	Input	GTY loopback port for test purpose. Default “000”.
USER_DATA_M_AXI_RX	Output	User Master AXI Stream port.
GT_SERIAL_TX	Output	Differential serial data output pin.
HARD_ERR	Output	Hard error detected (asserted until Aurora 8B/10B core resets).
SOFT_ERR	Output	Soft error detected in the incoming serial stream.
LANE_UP	Output	Asserted upon successful lane initialization.
mmcm_not_locked	Output	GTY Txpmaresetdone_out.
CHANNEL_UP	Output	Asserted when Aurora 8B/10B channel initialization is complete and the channel is ready for data transfer.
rx_resetdone	Output	GTY RX reset completed.
tx_resetdone	Output	GTY TX reset completed.
tx_lock	Output	GTY cpll lock.

Table 5: **Aurora 8B10B Ports List *(Cont'd)***

Port	Direction	Description
TRANSCEIVER_DEBUG	Input/Output	Additional Transceiver debugging ports.
user_clk_out	Output	GTY user clock out.

Test Bench Simulation

A behavioral simulation has been performed to validate the new Aurora core and a test bench is provided and tested with Vivado simulator and Mentor Questa Sim.

The demonstration test bench performs the following tasks:

- Generates input clock signals.
- Applies a reset to the example design.
- Waits for one of the interrupt signals (Link Status, Configuration Error (Aurora PHY) and Multi-Bit Error) to be asserted. If Link status is asserted, a stable link is established between the Master and Slave AXI Chip2Chip cores. If Configuration Error or Multi-Bit Error is asserted, the test bench fails with Error: Link Not Detected.
- If a link is successfully established, Link detected is displayed in the console.
- The traffic generator starts generating fixed traffic patterns at the inputs of the AXI Chip2Chip cores.
- The traffic checker checks the output signals of the AXI Chip2Chip cores against expected patterns. If the received data has an error, then error messages are issued at the console with the name, expected value and actual value of the signal in error condition.
- The transactions are shown for a time interval of 10,000 ns and the test bench finishes with the Test Completed Successfully in the console.

The following example shows a section of the log of the full simulation:

```
# Link detected
# R Match exp_axi_rdata  = ffffffff, exp_axi_rid    = 3f, exp_axi_rlast  = 0,
exp_axi_rresp  = 0
# R Match  s_axi_rdata  = ffffffff,  s_axi_rid   = 3f,  s_axi_rlast  = 0,  s_axi_rresp
= 0
# B Match exp_axi_bid   = 00, exp_axi_bresp = 3
# B Match   s_axi_bid   = 00,   s_axi_bresp = 3
#  W Match exp_axi_wdata = ffffffff, exp_axi_wuser   = f, exp_axi_wlast = 1,
exp_axi_wstrb = f
#  W Match  m_axi_wdata = ffffffff,  m_axi_wuser  = f,  m_axi_wlast = 1,  m_axi_wstrb
= f
# R Match exp_axi_rdata  = 00000002, exp_axi_rid    = 01, exp_axi_rlast  = 1,
exp_axi_rresp  = 1
# R Match  s_axi_rdata  = 00000002,  s_axi_rid   = 01,  s_axi_rlast  = 1,  s_axi_rresp
= 1
#  W Match exp_axi_wdata = fffffffe, exp_axi_wuser   = d, exp_axi_wlast = 0,
exp_axi_wstrb = f
#  W Match  m_axi_wdata = fffffffe,  m_axi_wuser  = d,  m_axi_wlast = 0,  m_axi_wstrb
= f
```

```
#  W Match exp_axi_wdata = ffffffd, exp_axi_wuser   = b, exp_axi_wlast = 1,
exp_axi_wstrb = f
#  W Match   m_axi_wdata = ffffffd,   m_axi_wuser   = b,   m_axi_wlast = 1,   m_axi_wstrb
= f
# AW Match exp_axi_awaddr = 00000000, exp_axi_awburst = 0, exp_axi_awid =
00,exp_axi_awlen = 00, exp_axi_awsize = 0
# AW Match   m_axi_awaddr = 00000000,   m_axi_awburst = 0,   m_axi_awid = 00, m_axi_awlen
= 00,   m_axi_awsize = 0
# M_AXI LITE AW channel exp aw 000000aa prot 2, act aw 000000aa prot 2
# M_AXI LITE W channel exp w 00000154 strb 4, act w 00000154 strb 4
# Test Completed Successfully
# ** Note: $finish    :
../../../../../../sources/testbench/axi_chip2chip_master_exdes_tb.v(133)
#    Time: 2675335100 ps  Iteration: 0  Instance: /axi_chip2chip_master_exdes_tb
```

The Figure 11 shows the waveform of the behavioral simulation. The pink signals are asserted when the link between master and slave works properly.

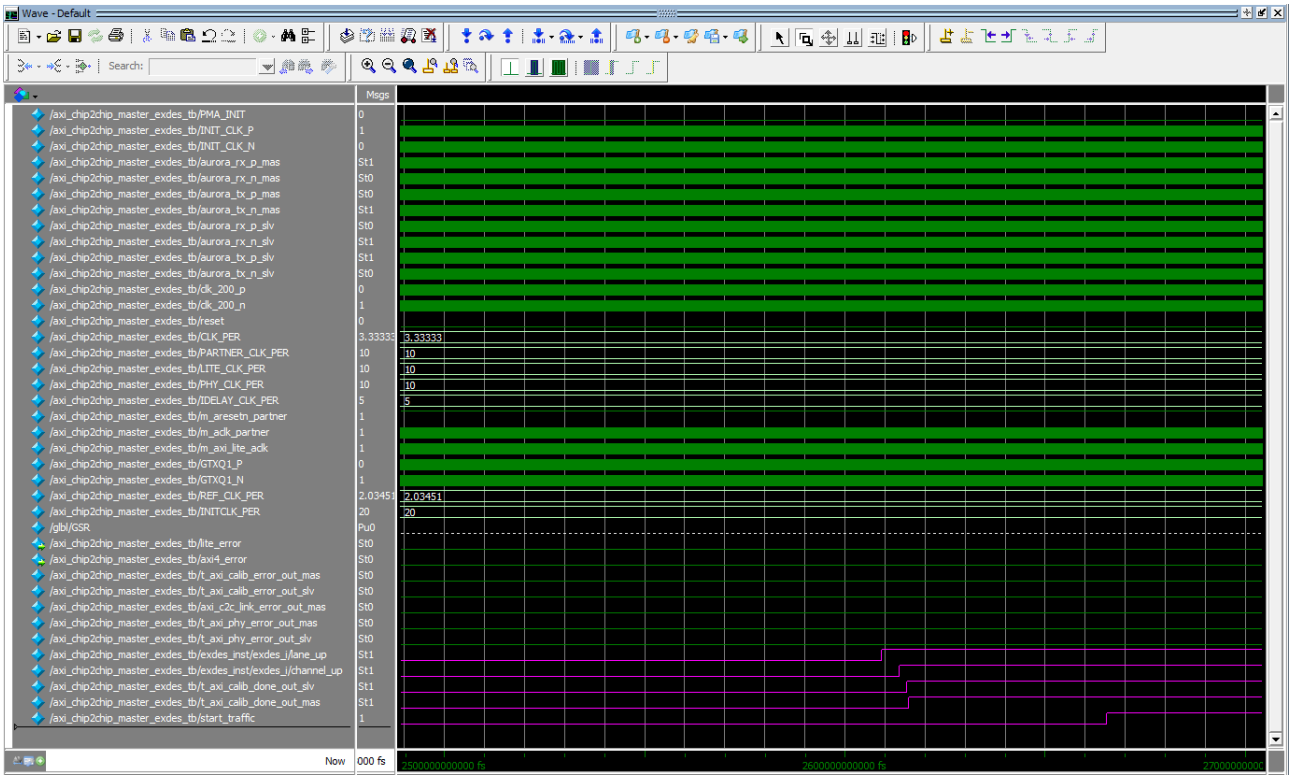

Figure 11: **Waveform Behavioral Simulation**

Reference Design Files

The Reference Design for the Master AXI Chip2Chip is for the Xilinx ZCU1254 and VCU1262 evaluation boards. The directory structure is the same for both. Figure 12 shows the directory structure for the Zynq® UltraScale+ RFSoC (ZCU1254) device design files.

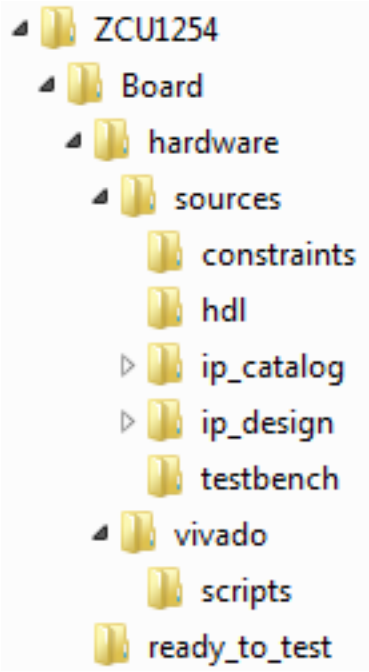

Figure 12: **ZCU1254 Directory Structure**

The ZCU1254 folder contains the hardware design deliverables listed in Table 6.

Table 6: **ZCU1254 Hardware Design Deliverables**

Folder Name	Description
Board	Contains all required scripts and a programming file for the board test
Sources/constraints	Contains the I/O and timing constraints file
Sources/hdl	Contains the source code deliverable files
Sources/ip_catalog	Contains the Xilinx IP cores required for the design
Sources/ip_design	Contains the Aurora_8B10B_gty IP and source files
Sources/testbench	Contains the test bench files for simulation
Vivado/scripts	Contains the design creation script for both Windows and Linux operating systems in command line and in Vivado design suite IDE mode
Ready_to_test	Contains programming files to configure the ZCU1254 evaluation board

The `readme.txt` file provides the details on the folder structure, tool version, and revision.

The Reference Design for the Slave AXI Chip2Chip is for the Xilinx AC701 evaluation. Figure 13 shows the directory structure for the Artix-7 device (AC701) design files.

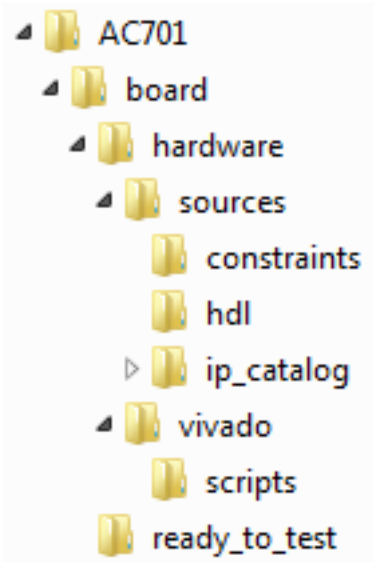

Figure 13: **AC701 Directory Structure**

The AC701 folder contains the hardware design deliverables listed in Table 7.

Table 7: **AC701 Hardware Design Deliverables**

Folder Name	Description
Board	Contains all required scripts and a programming file for the board test
Sources/constraints	Contains the I/O and timing constraints file
Sources/hdl	Contains the source code deliverable files
Sources/ip_catalog	Contains the Xilinx IP cores required for the design
Vivado/scripts	Contains the design creation script for both Windows and Linux operating systems in command line and in Vivado design suite IDE mode
Ready_to_test	Contains programming files to configure the AC701 evaluation board

The `readme.txt` file provides the details on the folder structure, tool version, and revision.

Project Creation

A script is provided in order to create the Vivado project. The script can be sourced either in a shell configured for the Vivado tool or in the Tcl console available in the GUI.

MASTER Reference Design

From the shell move to the scripts directory and execute the following command:

```
vivado -mode tcl -source ultrascalep_c2c_master.tcl
```

In the TCL Console of GUI, move to the script directory and type:

```
source ./ultrascalep_c2c_master.tcl
```

SLAVE Reference Design

From the shell move to the scripts directory and execute the following command:

```
vivado -mode tcl -source artix7_c2c_slave.tcl
```

In the TCL Console of GUI, move to the script directory and type:

```
source ./artix7_c2c_slave.tcl
```

The script creates the project by importing all of the design files needed for the design implementation and simulation.

Hardware Test

The reference design has been validated in hardware using the Xilinx Evaluation Board ZCU1254 rev. D, VCU1262 rev. B, and the AC701 rev1.0.

The location of the GTs can be fixed via xdc constraints. For both designs the constraints are delivered specific to the boards used during the development. If needed, they can be modified accordingly to the requirements.

Table 8 shows the Clock and Data connection for the boards used for the test.

Table 8: **Board Connections**

Board	Connector	Ref. Clock P/N	GT RX P/N	GT TX P/N
ZCU1254	Samtec BullsEye	128_CLK0	128_RX0	128_TX0
VCU1262	Samtec BullsEye	224_CLK0	224_RX0	224_TX0
AC701	SMA	J25/J26	J46/J47	J44/J45

The Vivado debugger is used to run the test. VIOs are defined to force the state of some signals. The Aurora link connection can also be tested individually looping the GTs. The loopback can be

done via external cable or with a VIO. "010" is a configuration for a Near-End PMA loopback. Details about loopback can be found in the *UltraScale Architecture GTY Transceivers User Guide* (UG578) [Ref 3] and *7 Series FPGAs GTP Transceivers User Guide* (UG482) [Ref 4].

Each board needs only an external reference clock of 491.52MHz. The other clocks needed for the reference designs are derived internally in the FPGAs from the reference clock.

Figure 14 shows the VIO setup of the Master design and the status of the outputs debug signals.

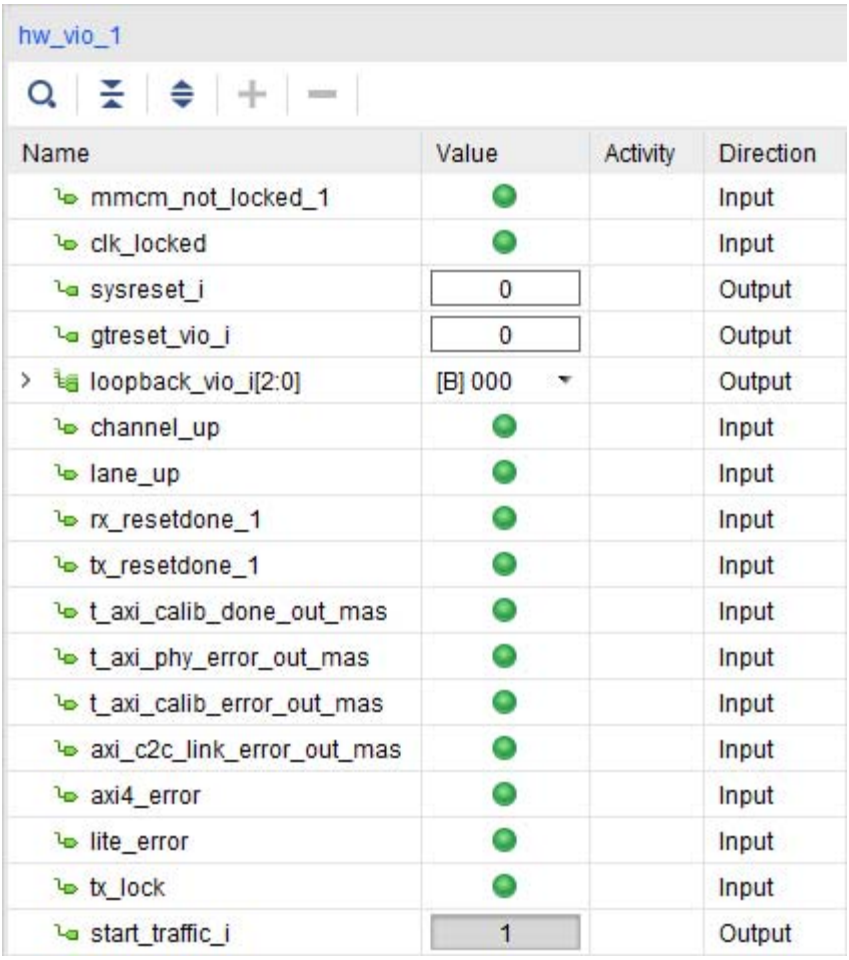

Figure 14: **Master VIOs Setup**

To run the test, use the VIO configuration in Table 9.

Table 9: **VIO Signals**

Signal	Description	Working Status Signal
Outputs		
sysreset_i	Design reset (not the Transceiver).	Active-High
gtreset_vio_i	Transceiver reset.	Active-High
start_traffic	The traffic generator starts generating fixed traffic patterns at the inputs of the AXI Chip2Chip cores.	Active-High

Table 9: **VIO Signals *(Cont'd)***

Signal	Description	Working Status Signal
loopback_vio_i[2:0]	Transceiver loopback	"000" Normal operation "001": Near-End PCS Loopback "010": Near-End PMA Loopback
Inputs		
mmcm_not_locked_1	Txpmaresetdone_out	0 or green
clk_locked	MMCM free running clock locked.	1 or green
lane_up	Asserted upon successful lane initialization.	1 or green
channel_up	Asserted when Aurora 8B/10B channel initialization is complete and the channel is ready for data transfer.	1 or green
rx_resetdone_1	GTY RX reset completed.	1 or green
tx_resetdone_1	GTY TX reset completed.	1 or green
t_axi_calib_done_out_mas (axi_c2c_link_status_out)	Link Status: Asserted when Link Detect FSM is in the SYNC state. Deasserted when either the Master or Slave AXI Chip2Chip core is under reset or when the Link Detect FSM is not in the SYNC state.	1 or green
t_axi_phy_error_out_mas (axi_c2c_multi_bit_error_out)	Multi-bit Error Interrupt: When asserted, this interrupt indicates multiple bits are received with errors in the Master or Slave AXI Chip2Chip core.	0 or green
t_axi_calib_error_out_mas	If the Master does not receive the expected pattern within a specified interval, it asserts the configuration error status signal.	0 or green
axi_c2c_link_error_out	Link Error Interrupt: Asserted when the AXI Chip2Chip Slave core is reset during normal operations. This signal is valid only in Master mode.	0 or green
axi4_error	Traffic checker error.	0 or green
lite_error	Traffic lite checker error.	0 or green
tx_lock	GTY cpll lock	1 or green

Figure 15 shows the VIO for the Slave design.

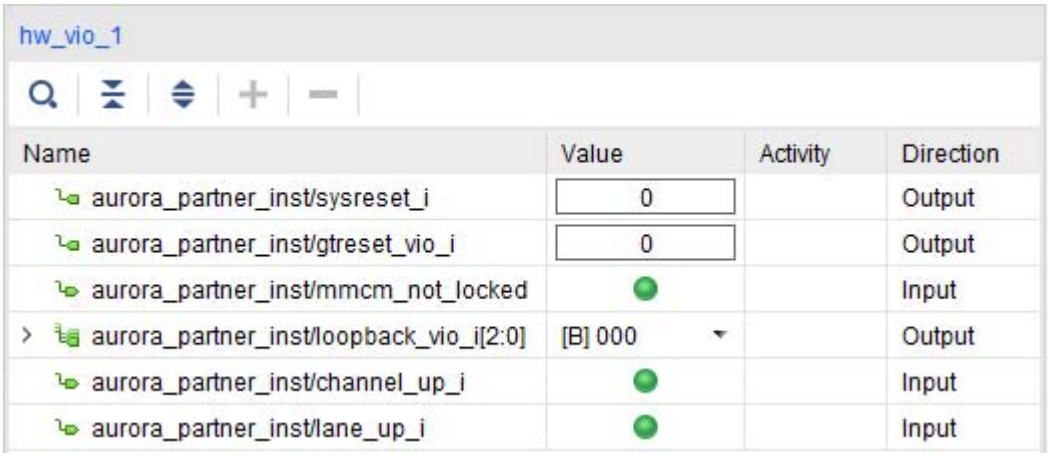

Figure 15: **Slave VIOs Setup**

To run the test, use the VIO configuration listed in Table 10.

Table 10: **VIO Signals**

Signal	Description	Working Status Signal
Inputs		
sysreset_i	Design reset (not the Transceiver).	Active-High
gtreset_vio_i	Transceiver reset.	Active-High
loopback_vio_i[2:0]	Transceiver loopback	"000" Normal operation "001": Near-End PCS Loopback "010": Near-End PMA Loopback
Outputs		
mmcm_not_locked	Txpmaresetdone_out	0 or green
channel_up_i	Asserted when Aurora 8B/10B channel initialization is complete and the channel is ready for data transfer.	1 or green
lane_up_i	Asserted upon successful lane initialization.	1 or green

Figure 16 shows the two boards used for the test validation.

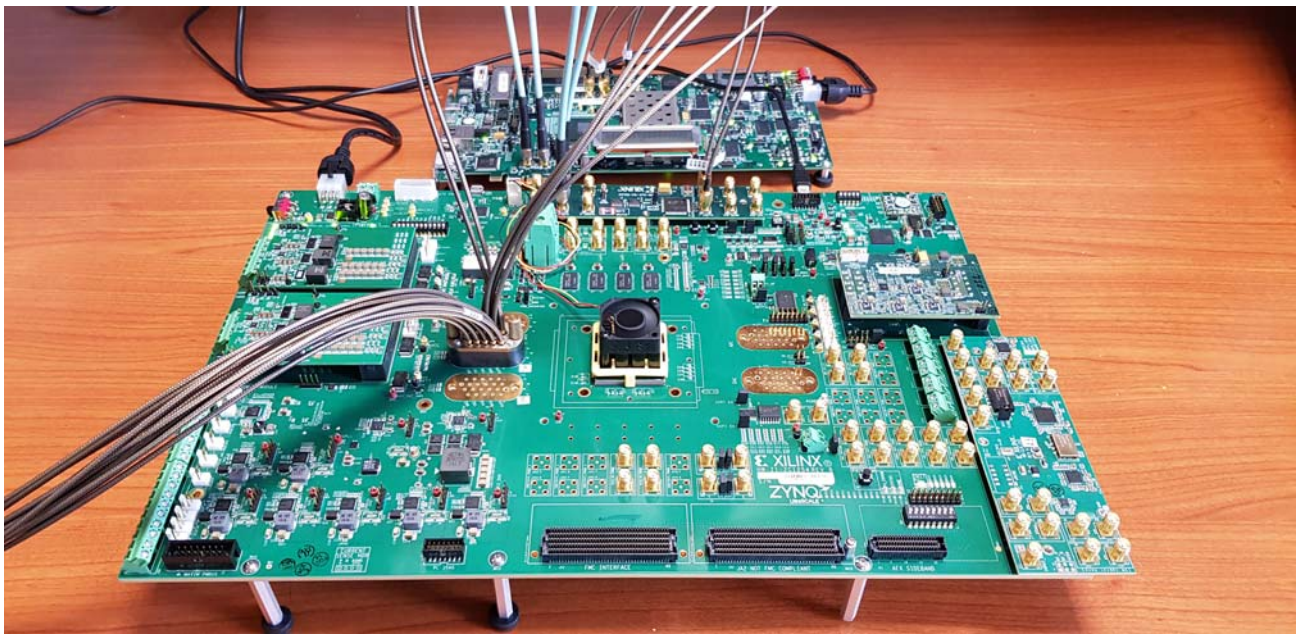

Figure 16: **ZCU1254 and AC701 Boards**

Conclusion

The reference design has been developed with the following features:

- Interface
  - Master RFSoC: AXI Chip2Chip interface, Aurora 8B10B 1 lane @1.2288Gbit/s
  - Slave A7: AXI Chip2Chip interface, Aurora 8B10B 1 lane @1.2288Gbit/s
- Data connectivity to user application is through AXI Lite interfaces.
- The reference design includes the test logic to validate the new Aurora 8B10B IP.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

References

1. *AXI Chip2Chip LogiCORE Product Guide* (PG067)
2. *Aurora 8B/10B LogiCORE Product Guide* (PG046)
3. *UltraScale Architecture GTY Transceivers User Guide* (UG578)
4. *7 Series FPGAs GTP Transceivers User Guide* (UG482)
5. *KCU105 Evaluation Board User Guide* (UG917)
6. *AC701 Evaluation Board for the Artix-7 FPGA User Guide* (UG952)

Revision History

The following table shows the revision history for this document.

Date	Version	Changes
06/06/2018	1.0	Initial Xilinx release.

Please Read: Important Legal Notices